♈ ♉ ♊ ♋ ♌ ♍ ♎ ♏ ♐ ♑ ♒ ♓

How many star signs actually are there?

By Adam Yates

Introduction

Your star sign is the zodiac constellation the sun was in at the moment of your birth, some say it represents your identity, personality and sense of 'self' in astrology although there is minimal scientific proof of this... The Babylonians suggested there are exclusively 12 star signs, each corresponding to a constellation. However, the Yale bright star catalogue contains data about 9110 stars, visible to the naked eye, at night. Therefore, if each zodiac constellation has approximately 10 stars. We should expect ~9000 stars to be left over that could all form their own star signs. So, it is only safe to assume that there are more star signs in the night sky than the Babylonians first thought, right?

First thoughts

Firstly, lets define what makes up constellation, For our example we will take the smallest 'group' of stars observed in one of the 12 star signs as our minimum and the largest 'group' as our maximum. So Aries, which has a minimum group size of 4 stars and Pisces with a group size of 18 stars (htt4). So, mathematically, a constellation is the connection of 4-18 stars using line segments where the stars are vertices. So, to count the maximum number of possible constellations, surely, we can take the 9110 visible stars and put them in the smallest possible group sizes, 4. And conclude that,

$$9110 \div 4 = 2277 \; constellations$$

Wrong! This is an underestimate of the total **possible** number of constellations. This is because, in the previous example, it's assumed that when one star is 'used' in one constellation it can't be used again in another. To calculate all the possible 4-star-sized constellations in the sky we will have to use the combination function, which will calculate all the possible arrangements.

$$^{9110}C_4 = \frac{9110!}{4!\,(9110-4)!} = \frac{1 \times 2 \times \; \dots \; \times \; 9110}{1 \times 2 \times \; \dots \; \times \; 9106} \div 4! = \frac{9110 \; \times \; 9109 \; \times 9108 \; \times 9107}{4!} \; = \; 2.87 \times 10^{14}$$

But this number lacks serious logic for two reasons. What about the constellations which consist of more than 4 stars? And can any star be connected to any other star? Firstly, of course, constellations can be made up of more than four stars, so this needs to be taken into consideration. And secondly, no, this is because, in practice, constellations consist of stars which are close together in the sky, not any star spread across the entire sky. But to analyse which stars can be used to create constellations with each other we will have to refer back to the Yale bright star catalogue.

What does the sky even look like?

HR	Name	RAh	RAm	RAs	DE-	DEd	DEm	DEs	Vmag	pmRA	pmDE
1		0	5	9.9	+	45	13	45	6.7	-0.012	-0.018
2		0	5	3.8	-	0	30	11	6.29	0.045	-0.06
3	33 Psc	0	5	20.1	-	5	42	27	4.61	-0.009	0.089
4	86 Peg	0	5	42	+	13	23	46	5.51	0.045	-0.012
5		0	6	16	+	58	26	12	5.96	0.263	0.03
6		0	6	19	-	49	4	30	5.7	0.565	-0.038
7	10 Cas	0	6	26.5	+	64	11	46	5.59	0.008	0
8		0	6	36.8	+	29	1	17	6.13	0.38	-0.182
9		0	6	50.1	-	23	6	27	6.18	0.1	-0.045
10		0	7	18.2	-	17	23	11	6.19	-0.018	0.036
11		0	7	44.1	-	2	32	56	6.43	0.027	-0.002
12		0	7	46.8	-	22	30	32	5.94	0.052	-0.044
13		0	8	3.5	-	33	31	46	5.68	-0.037	0
14		0	8	12.1	-	2	26	52	6.07	0.009	-0.003
15	21Alp And	0	8	23.3	+	29	5	26	2.06	0.136	-0.163
16		0	8	17.4	-	8	49	26	5.99	-0.052	-0.033
17		0	8	41	+	36	37	36	6.19	-0.1	-0.145
18		0	8	33.4	-	17	34	39	6.06	0	-0.025
19		0	8	52.2	+	25	27	46	6.23	0.114	0.031
20		0	9	20.2	+	79	42	53	6.01	0.102	-0.027
21	11Bet Cas	0	9	10.7	+	59	8	59	2.27	0.525	-0.181

After abstracting unnecessary data from the catalogue, we are left with 12 relevant columns telling us about the stars and their positions in our sky. But only 7 of these refer to the stars' positions in the sky. RAh, RAm and RAs refer to the right ascension of the stars. This is the angle which measures how far east a star is from a fixed point in the sky, it is measured in hours, minutes and seconds rather than typical degrees to reflect the sky 'rotating' once every 24 hours. Similarly, DE-, DEd, DEm and DEs are columns related to the declination and measures how north or south a star is from a fixed point, it is measured in degrees, minutes and seconds with its polarity or direction in a separate column.

For parity, convert all the right ascension and declination values into single degree values using the formulae,

$$RA^\circ = 360 \times \left(\frac{RA_{hours}}{24} + \frac{RA_{minutes}}{24 \times 60} + \frac{RA_{seconds}}{24 \times 60 \times 60}\right)$$

$$DE^\circ = DE_{polarity} \times \left(DE_{degrees} + \frac{DE_{minutes}}{60} + \frac{DE_{seconds}}{60 \times 60}\right)$$

Secondly, create a blank image in python. This will create a grid of pixels each with a coordinate we can use to draw our stars on plotting them using their position:

$$(RA^\circ, DE^\circ)$$

If this were a typical coordinate plane this would work fine however, this python module defines the coordinates of the pixels differently, and additionally we want to have extra pixels to allow for stars with close coordinates to be separate. The coordinate grid increases x value as you move right, as usual, but increases y value as you move downwards, with the origin being in the top left corner. If we let,

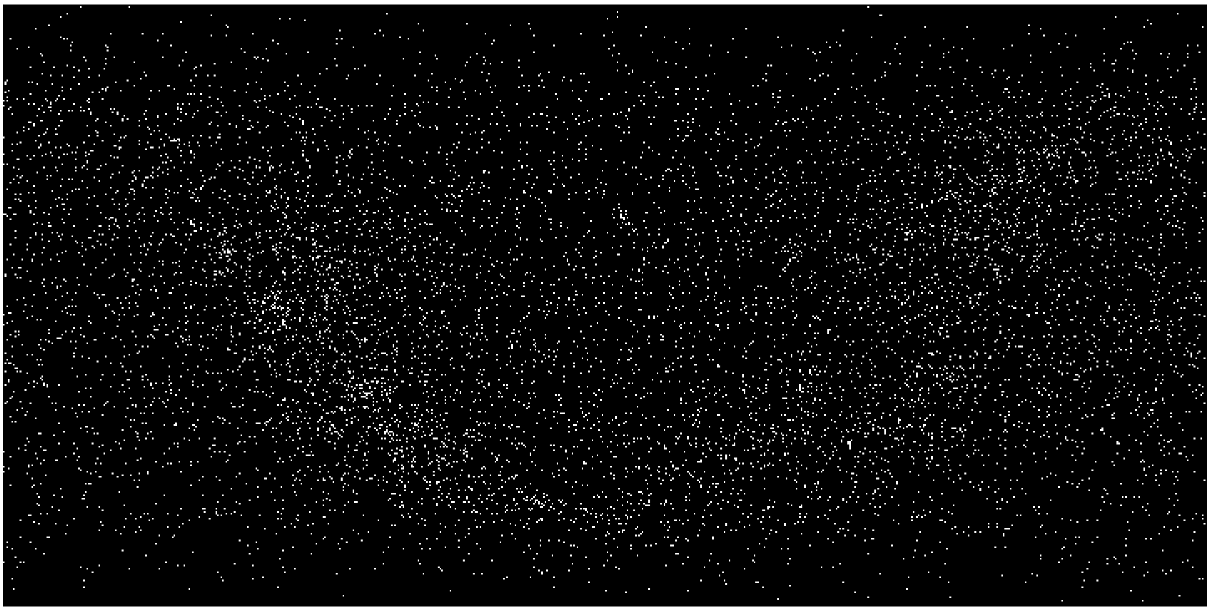
$$H = height\ of\ image$$

$$W = width\ of\ image$$

then the new coordinates will be,

$$\left((W-1) \times \frac{RA°}{360}, (H-1) \times \frac{(90 - DE°)}{180}\right)$$

and plotting these results in the following scatter graph,



By plotting the stars on axes that are both measured in degrees, we have plotted them on an equirectangular projection, it's a similar issue to how cartographers tried to project a sphere onto a rectangular map, there is distortion. However, a notable observation of this graph is the slightly denser curve of stars. Imagine wrapping this image around the Earth, the denser curve of stars would actually 'flatten out' and form a straight line. What this demonstrates is that the distribution of the stars in the sky is not uniform and so a more complex way of grouping nearby stars is needed...

Circles on spheres

To group stars on this map it may be tempting to draw circles on the grid. But as we discussed, this would simple not work, they would have been distorted. We need the equation of a circle on a sphere, or, a spherical circle.

A circle is created on a sphere when the angle between the centre of the sphere and any given point are equal to some constant, so, start with the parametric equation of a unit sphere. This is the equation of a sphere based off right ascension ($RA°$) and declination ($DE°$).

$$x = \cos(DE°)\cos(RA°)$$

$$y = \cos(DE°)\sin(RA°)$$

$$z = \sin(DE°)$$

Now, express a point on the sphere as a vector,

$$\boldsymbol{u} = \begin{pmatrix} \cos(DE^\circ)\cos(RA^\circ) \\ \cos(DE^\circ)\sin(RA^\circ) \\ \sin(DE^\circ) \end{pmatrix}$$

To find the angle between two points on the circle use the dot product between two points, so,

$$\boldsymbol{u_1} \cdot \boldsymbol{u_2} = |\boldsymbol{u_1}||\boldsymbol{u_2}|\cos(r)$$

However, since this is a unit sphere, the magnitude of the vector between the centre of the sphere and any given point is the radius so it is 1. Therefore,

$$\boldsymbol{u_1} \cdot \boldsymbol{u_2} = \cos(r)$$

Now calculate the dot product
(Note that $DE^\circ{}_n$ and $RA^\circ{}_n$ are used to denote the angles of the nth point),

$$\cos(DE^\circ{}_1)\cos(RA^\circ{}_1)\cos(DE^\circ{}_2)\cos(RA^\circ{}_2) + \cos(DE^\circ{}_1)\sin(RA^\circ{}_1)\cos(DE^\circ{}_1)\sin(RA^\circ{}_1) + \sin(DE^\circ{}_1)\sin(DE^\circ{}_2) = \cos(r)$$

And we are left with a monster of an equation, so factorise to give,

$$\cos(DE^\circ{}_1)\cos(DE^\circ{}_2)\left[\cos(RA^\circ{}_1)\cos(RA^\circ{}_2) + \sin(RA^\circ{}_1)\sin(RA^\circ{}_2)\right] + \sin(DE^\circ{}_1)\sin(DE^\circ{}_2) = \cos(r)$$

And we are still left with something slightly monstrous looking! So, use the cosine subtraction identity,

$$\cos(\mathrm{A})\cos(B) + \sin(A)\sin(B) \equiv \cos(A - B)$$

Which gives us neatly,

$$\cos(DE^\circ{}_1)\cos(DE^\circ{}_2)\cos\,(RA^\circ{}_1 - RA^\circ{}_2) + \sin(DE^\circ{}_1)\sin(DE^\circ{}_2) = \cos(r)$$

However, since our axis are measured in RA° and DE°, we can replace with x and y to give us:

$$\cos(r) = \sin(y)\sin(DE^\circ) + \cos(y)\cos(DE^\circ)\cos\,(x - RA^\circ)$$

Where (RA°, DE°) is the centre of our circle.

You can see how this works using the Desmos graph I have created here.

This formula for measuring angular distance will be important later…

How big are star signs?

To find the maximum angular size of a star sign, we must look at the largest zodiac constellation Pisces, which uniquely contains a long 'cord' in its structure and thus, the greatest angular distance between two stars within its constellation amongst all the zodiac signs.

The two stars are called tau piscium and gamma piscium (htt1)and using their right ascension and declination values from the Yale bright star catalogue we can calculate their angular distance by substituting into our original formula.

$$\cos(30.09^\circ)\cos(3.28^\circ)\cos\,(17.92^\circ - 349.29^\circ) + \sin(30.09^\circ)\sin(3.28^\circ) = \cos(r)$$

$$\cos(r) = 0.78688 \ldots$$

$$r \approx 51.9^\circ$$

This is an important value and will be used in our final calculations.

Sun contact

All the star signs are constellations that lie along the ecliptic, so from the Earth's view the Sun passes through them once a year. But our diagram of the night sky consists of 9110 stars where this property may not be true. To narrow it down to stars which actually have this essential star sign property we need to look at the path of the Sun.

Every year, the Sun's right ascension increases by 360 degrees, and assuming our Earth orbits the Sun in an approximate circular shape. Then,

$$RA° = 360 \times \frac{t}{365}$$

Where $RA°$ is the right ascension of the sun and t is the days since the start of the year.

To model the Sun's declination, we need to consider the Earth's tilt. This causes the Sun, relative to us, to oscillate with an amplitude of $23.44°$ (the Earth's tilt). So,

$$DE° = 23.44 \times \sin(RA°)$$

But considering the angular diameter of the sun as $0.545°$ or angular radius of $\frac{0.545°}{2} = 0.2725°$. We can calculate the 'upper bound' and 'lower bound' of our declination. (No, no! I hear you cry, as this is **obviously** ignoring distortion of the projection mentioned earlier. But this is for simplicity and regardless, this will make minimal difference to the results as the declination is quite small, where distortion happens the least).

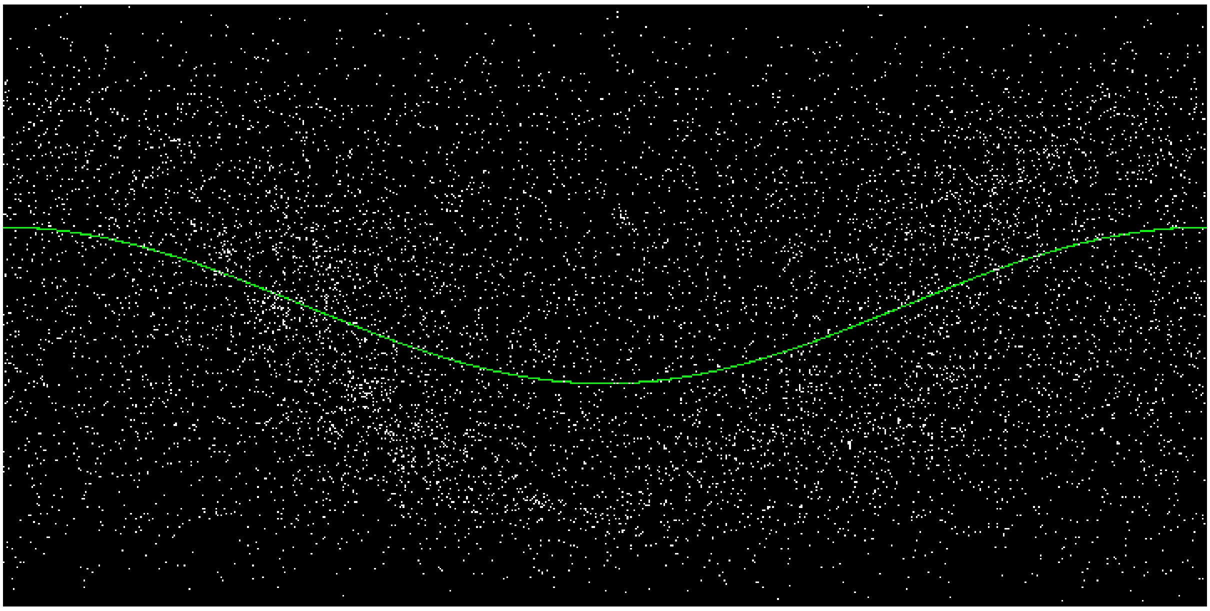
So, our declination is now given by,

$$DE° \pm 0.2725° = 23.44 \times \sin(RA°)$$

So, the Sun will pass through all the stars which satisfy the inequality,

$$23.44 \times \sin(RA°) - 0.2725 \leq DE° \leq 23.44 \times \sin(RA°) + 0.2725$$

And thus, we can check all 9110 stars as to whether they satisfy these equations and reject the ones that don't.



Plotting this on our map shows that only a small number of stars meet this criteria and counting them (on a much higher scale image to negate rounding errors) reveals that there are only 39 stars in this inequality.

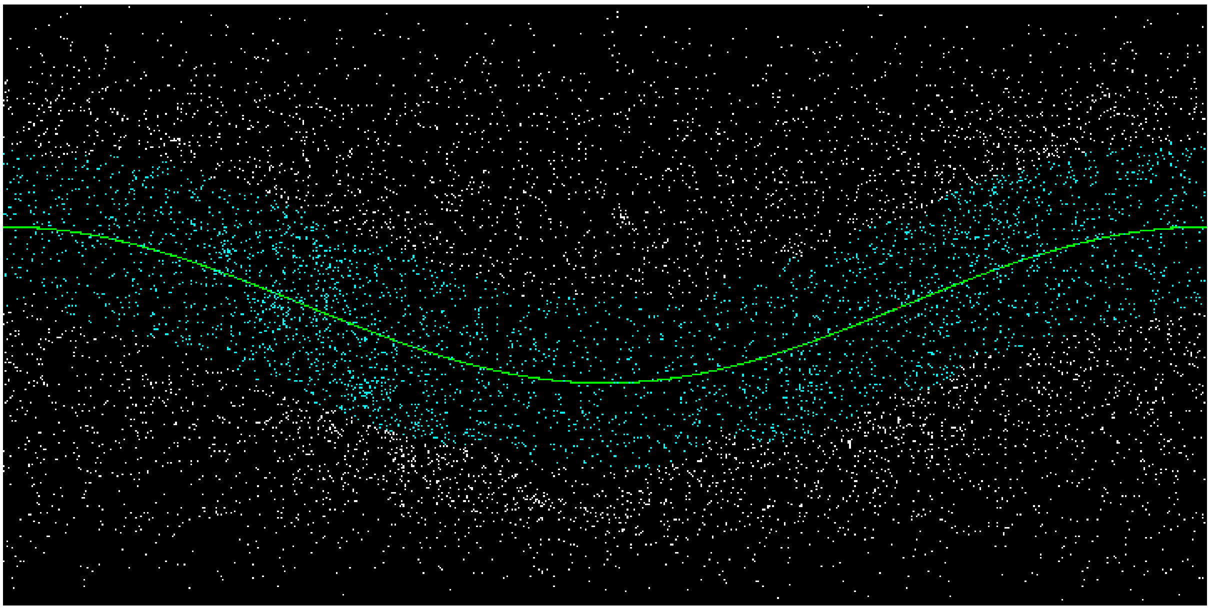
From this we can say that a star sign constellation should be roughly centred around one of these 39 stars which I will be calling a 'permitted star'

Final calculations

Finally, we can establish the real rules for a constellation to be deemed as a zodiac star sign.

1. There are between 4 and 18 stars in the constellation
2. The maximum angular size of the constellation is 51.9°
3. The constellation should be roughly centred around a star that the sun passes through.

To satisfy rule's 2 and 3, for each permitted star we should count the number of stars within $\frac{51.9°}{2}$ angular distance of it. This will ensure that the constellation will be roughly centred around the star. We can calculate this by using our angular distance formula between each permitted star every other star. Luckily, the 21st century lets us do this very quickly using the help of Python.



The program outputs a list on numbers, each number corresponds to the number of nearby stars that a constellation can be made with. The diagram has also been updated to highlight these stars in blue.

To calculate all the combinations for each permitted star lets define a function,

$$f(n) = nC4 + nC5 + \cdots + nC18$$

Where n is the number of nearby stars.

But even then, those are just the combinations of stars. For each group with S stars there are $SC2$ different possible line segments that can be drawn to connect them. Each line segment, in a constellation, either exists or it doesn't. So, there are 2 states a line segment can be in. Therefore, there are 2^{SC2} possible ways to draw a constellation given S stars. And, to count the number of drawable constellations we should actually have,

$$f(n) = nC4(2^{4C2}) + nC5(2^{5C2}) + \cdots + nC18(2^{18C2})$$

This number is too huge to calculate with simple calculators, especially when n is in the hundreds. However, we can take our largest term which will dominate the rest of the terms to get a solid estimate. So,

$$f(n) \sim nC18(2^{18C2}) = 2^{153} \times nC18$$

Now we must calculate f(n) for all of the nearby stars we previously calculated. And sum them. So our final answer should be,

$$number\ of\ star\ signs = \sum f(\boldsymbol{n}) = \sum 2^{153} \times \boldsymbol{n}C18 = 2^{153} \times \sum \boldsymbol{n}C18$$

If we loop through all values of n in Python, we should be able to simplify this further, and we are given the value,

```
17351482334779111340101969840882 4399
10^ 81.14157110170393
```

So,

$$number\ of\ star\ signs \approx 2^{153} \times 10^{81.14}$$

Change base of 2 to match 10,

$$2^{153} = 10^{k}$$

$$\log(2^{153}) = \log(10^{k})$$

$$153\log(2) = k$$

$$k \approx 46.06$$

So,

$$number\ of\ star\ signs \approx 10^{46.06} \times 10^{81.14}$$

And finally,

$$number\ of\ star\ signs \approx 10^{127.2}$$

Conclusion

While this number approximates how many possible star signs could have been created, it does raise some interesting questions. What factors decided that the 12 star signs in astrology were selected? Was it luminosity, artistic value or just by chance? But it does tell us that there is at least one star sign for each of the $10^{9.917}$ humans on Earth. So maybe astrology was right all along, and each person's identity, personality and sense of 'self' was determined by a uniquely possible star sign.

(1986 words)

Works Cited

(n.d.). Retrieved from https://en.wikipedia.org/wiki/Sun

(n.d.). Retrieved from https://en.wikipedia.org/wiki/Gamma_Piscium

(n.d.). Retrieved from https://en.wikipedia.org/wiki/Tau_Piscium

(n.d.). Retrieved from https://www.constellation-guide.com/constellation-list/pisces-constellation),

(n.d.). Retrieved from https://en.wikipedia.org/wiki/Pisces_(constellation)

(n.d.). Retrieved from https://en.wikipedia.org/wiki/Aries_(constellation)

worldometer. (n.d.). Retrieved from https://www.worldometers.info/world-population/

Appendix

Appendix A: Python Implementation (not counted to word count)

```
import pandas as pd
import numpy as np
from PIL import Image
import math

# Image size
WIDTH = 360*2
HEIGHT = 180*2

# Load only relevant data
df = pd.read_csv(
    "stars.csv",
    usecols=[10 ,14, 15],
    skiprows=1,
    nrows=9110
)

df = df.apply(pd.to_numeric, errors='coerce')
df = df.dropna()

RA = df.iloc[:, 1].to_numpy()
DE = df.iloc[:, 2].to_numpy()

# Normalise coordinates

x = (RA / 360.0) * (WIDTH - 1)
y = ((90 - DE) / 180.0) * (HEIGHT - 1)

# Convert to integer indices
x = x.astype(np.int32)
y = y.astype(np.int32)

# Create empty sky
img = Image.new("RGB", (WIDTH, HEIGHT), "black")
pixels = img.load()

# Draw points
coords = set(zip(x, y))

sunPassThroughStars = []
for xi in range(0,WIDTH):
    for yi in range(0,HEIGHT):

        xval = (xi / (WIDTH - 1)) * 360
        yval = -1*((((yi)/(HEIGHT - 1))*180)-90)

        if (xi, yi) in coords:
            pixels[xi, yi] = (255, 255, 255)
        if yval <= 23.44*np.cos(xval * (3.1415/180)) + 0.2725:
            if yval >= 23.44*np.cos(xval * (3.1415/180)) - 0.2725:
                pixels[xi, yi] = (0, 255, 0)
                if (xi, yi) in coords:
                    sunPassThroughStars.append((xval,yval))
                    pixels[xi, yi] = (255, 0, 0)

def getAngularDistance(star1, star2):
    DE1 = star1[1] * (3.1415/180)
    RA1 = star1[0] * (3.1415/180)
    DE2 = star2[1] * (3.1415/180)
    RA2 = star2[0] * (3.1415/180)

    angularDistanceRadians = np.arccos(np.cos(DE1)*np.cos(DE2)*np.cos(RA1 - RA2) + np.sin(DE1)*np.sin(DE2))

    return angularDistanceRadians/(3.1415/180)

NearbyNums = []
for selectedStar in sunPassThroughStars:
    totalNearby = 0
    for indexStar in coords:

        indexStarX = (indexStar[0] / (WIDTH - 1)) * 360
        indexStarY = -1 * (((indexStar[1] / (HEIGHT - 1)) * 180) - 90)

        if getAngularDistance(selectedStar, (indexStarX, indexStarY)) < (51.9 / 2):
            pixels[indexStar[0], indexStar[1]] = (0, 255, 255)
```

```
            totalNearby += 1
    NearbyNums.append(totalNearby)

# Save
img.save("star_map.png")
OpenImage = Image.open("star_map.png")
OpenImage.show()

totalConstellations = 0
for val in NearbyNums:
    totalConstellations += math.comb(val,18)
print(totalConstellations)
print("10^",str(math.log(totalConstellations)))

print("Done!")
```